05

Introducing ROS

All the high-level considerations of Part I have to be instantiated *somehow*. How do we keep track of the state of robot? Implement a controller? Communicate among robots? Interface with a user? The answer is almost always: (with difficulty and) with *computers*. And we know what that means: *software*.

computers
software

As we saw in Part I, *robots are complicated.* Can you imagine the amount of software required to run a given robot? A *ton*. Not to mention the expertise in several sub-fields within robotics. In the late 2000s, roboticists started the difficult but important task of collaboratively developing an open-source software framework that can be used to program many different types of robots: the *Robot Operating System* (ROS) (Quigley et al., 2009).

ROS

ROS is a framework in that it brings together code libraries, code tools, and development conventions to create a system in which individual applications can be developed. Many robotics researches share their expertise and development work via this framework, which means (among other things) cutting-edge libraries are available to everyone.

We adopt this platform, which is now ubiquitous.